WarehouseSyncSchedule.apxc :-

```
global class WarehouseSyncSchedule implements Schedulable {
    global void execute(SchedulableContext ctx) {

        WarehouseCalloutService.runWarehouseEquipmentSync();
    }
}
```

WarehouseSyncScheduleTest.apxc :-

```
@isTest

public class WarehouseSyncScheduleTest {



@isTest static void testScheduler() {

Test.SetMock(HttpCallOutMock.class, new WarehouseCalloutServiceMock());



String CRON_EXP = '0 0 0 1 1/1 ? *'; // To be executed monthly at day one

Integer runDate = 1;



DateTime firstRunTime = System.now();

DateTime nextDateTime;



if(firstRunTime.day() < runDate) {

nextDateTime = firstRunTime;

} else {

nextDateTime = firstRunTime.addMonths(1);

}



Datetime nextRunTime = Datetime.newInstance(nextDateTime.year(),
nextDateTime.month(), runDate);
```

```
Test.startTest();

WarehouseSyncSchedule warehouseSyncSchedule = new WarehouseSyncSchedule();


String jobId = System.schedule('Test Scheduler',

CRON_EXP,

warehouseSyncSchedule);


Test.stopTest();


// Get the information from the CronTrigger API object

CronTrigger ct = [SELECT Id, CronExpression, TimesTriggered, NextFireTime FROM
CronTrigger WHERE Id = :jobId];


// Verify the expressions are the same

System.assertEquals(CRON_EXP, ct.CronExpression);


// Verify the job has not run

System.assertEquals(0, ct.TimesTriggered);


// Verify the next time the job will run

System.assertEquals(String.valueOf(nextRunTime), String.valueOf(ct.NextFireTime));


}


}
```